Course Title: **Compilers and Languages** | Course Code: CCE3113 3rd year
Date: 14.11.2016 (First term) | Allowed time: 1 hrs

Answer the following questions:

Question No. 1 **(20 marks)**

For each of the following choose the letter introducing the best answer. (Check all that apply.) Explain your answer. (Each one is worth two degrees. The explanation is worth one degree.)

1. How many strings of length less than 4 are in the language described by the regular expression (x + y)* y (a + ab)*?
   a) 7
   b) 10
   c) 12
   d) 11
2. Regular expression for all strings starts with ab and ends with bba is.
   a) aba*b*bba
   b) ab(ab)*bba
   c) ab(a+b)*bba
   d) All of the mentioned
3. Following context free grammar
   S ⟶ aB | bA
   A ⟶ a | aS | bAA
   B ⟶ b | bS | aBB
   generates strings of terminals that have
   a) equal number of a's and b's
   b) odd number of a's and odd number b's
   c) even number of a's and even number of b's
   d) odd number of a's and even number of b's
4. The CFG
   S ⟶ aS | bS | a | b is equivalent to regular expression
   a) (a + b)
   b) (a + b) (a + b)*
   c) (a + b) (a + b)
   d) None of these
5. Any string of terminals that can be generated by the following CFG is
   S ⟶ XY
   X ⟶ aX | bX | a
   Y ⟶ Ya | Yb | a
   a) has at least one 'b'
   b) should end with 'a'
   c) has no consecutive a's or b's
   d) has at least two a's
6. For the following two regular languages **L1 = (a + b)* a** and **L2 = b (a + b)** *, the intersection of L1 and L2 is given by:
   a) (a + b) * ab
   b) ab (a + b) *
   c) a (a + b) * b
   d) b (a + b) * a

7. Which of the following grammars are ambiguous?

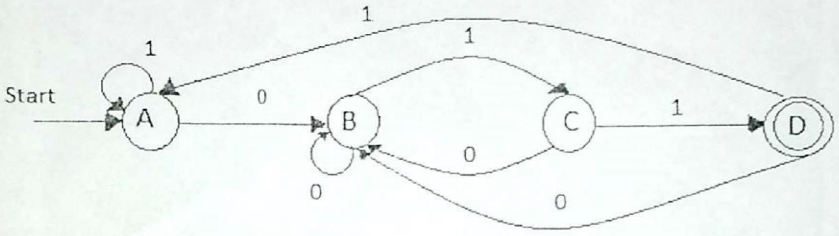
a) S → aSb | AA
   A → c | S

b) S → aSbS | aS | c

c) S → AaA | AbA
   A → c | S

d) S → aSbc | AB
   A → a
   B → a

8. In the following figure a deterministic finite automation M, Which of the following regular expressions denoted the set of all words accepted by M?

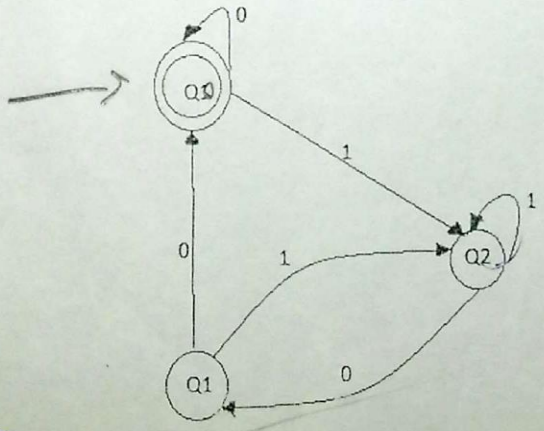


a) 001
b) 10*1*0
c) (0 | 1) * 011
d) 1* 0 * 00

9. Finite state machines _____ recognize nested structures

a) can
b) can't
c) may
d) may not

10. Regular expression corresponding to the state diagram given in the following figure is:



a) (0+1(1 + 01)* 00)*
b) (1 + 0 (0 + 10) 00)*
c) (0 + 1 (1 + 10) 00)*
d) (1 + 0(1 + 00) 11)*

Question No. 2 **(6 marks)**

1. **Explain why the grammar below is ambiguous.** (2 marks)

$S \rightarrow 0A \mid 1B$

$A \rightarrow 0AA \mid 1S \mid 1$

$B \rightarrow 1BB \mid 0S \mid 0$

2. **Consider the following grammar with terminals T = {a, b}.**

$S \rightarrow A\,\alpha \mid \delta$

$A \rightarrow S\,\beta$

(a) Can this grammar be recognized by a recursive descent parser? Why or why not? (2 marks)

(b) If not, How can you rewrite this grammar to make it a recursive descent grammar? (2 marks)

Question No. 3 **(4 marks)**

Consider the regular expression below which can be used as part of a specification of the definition of exponents in floating-point numbers. Assume that the alphabet consists of numeric digits ('0' through '9') and alphanumeric characters ('a' through 'z' and 'A' through 'Z') with the addition of a selected small set of punctuation and special characters (say in this example only the characters '+' and '-' are relevant). Also, in this representation of regular expressions the character '.' denotes concatenation.

$$\text{Exponent} = (+ \mid - \mid \varepsilon)\,.\,(E \mid e)\,.\,(\text{digit})+$$

For this regular expression, derive the DFA that is able to recognize this language.

Best wishes

Dr. Sherin El Gokhy